



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/904,989	07/13/2001	Neil A. Cooper	ATI.0100820	3444
34456 7590 06/21/2007 LARSON NEWMAN ABEL POLANSKY & WHITE, LLP 5914 WEST COURTYARD DRIVE SUITE 200 AUSTIN, TX 78730			EXAMINER CAO, DIEM K	
			ART UNIT 2194	PAPER NUMBER
			MAIL DATE 06/21/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

MAILED

JUN 21 2007

Technology Center 2100

Application Number: 09/904,989
Filing Date: July 13, 2001
Appellant(s): COOPER, NEIL A.

Adam D. Sheenhan
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/12/2007 appealing from the Office action
mailed 6/5/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is incorrect. A correct statement of the status of the claims is as follows:

Claims 2, 12 and 22 been canceled.

Claims 1, 3-11, 13-21 and 23-36 are rejected under 35 U.S.C. § 103.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is substantially correct. The changes are as follows: Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bondy et al. (U.S. 5,491,813) in view of Keller et al. (U.S. 5,752,032) and Schoening et al. (U.S. 6,226,788) further in view of Shirakabe et al. (U.S. 5,136,709).

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,491,813	BONDY ET AL.	2-1996
5,752,032	KELLER ET AL.	5-1998
6,226,788 B1	SCHOENING ET AL.	5-2001
5,136,709	SHIRAKABE ET AL.	8-1992

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1, 13 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bondy et al. (U.S. 5,491,813) in view of Keller et al. (U.S. 5,752,032) further in view of Schoening et al. (U.S. 6,226,788).

As to claim 1, Bondy teaches loading device-independent driver code (graphic packages 56, 57, 58, col. 6, lines 7-17), wherein the device-independent code forms a first portion of a

display driver (code which interacts with applications 51, 52, 53; col. 4, lines 27-42), receiving a device identifier associated with a particular device (Silicon Graphics, graPHIGS, col.4, lines 55-58), identifying a particular device-specific driver portion (device specific code 81 or 82) from a plurality of driver portions associated with the device identifier (col. 4, lines 27-42), loading the particular device-specific portion (col. 6, lines 18-30 and 46-53), wherein the device-specific portion forms a second portion of the display driver (code which interact with display adapter A, B, ..., E, Figs. 1,2). See col. 2, lines 11-53; col. 4, line 18 – col. 5, line 45; col. 9, line 41 – col. 10, line 16.

However, Bondy does not explicitly teach loading the device-independent driver code and the particular device-specific driver portion into kernel mode memory, and requesting a device identifier after loading the device-independent code into kernel memory, wherein the requested device identifier is to identify a particular device, and the identifying step is based on a comparison of versions associated with functions of the device-specific driver portion to versions expected through an application program interface. Keller teaches loading the device-independent driver code and the particular device-specific driver portion into kernel mode memory (kernel memory; col. 7, line 61 – col. 8, line 14), and requesting a device identifier after loading the device-independent code into kernel memory, wherein the requested device identifier is to identify a particular device (board identifier; col. 13, lines 5-20). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Bondy and Keller because it provides a flexible, modular device driver architecture that can provide independent hardware configuration options on a dynamic reconfiguration basis (col. 3, lines 14-17).

Schoening teaches device driver management, including locating a name associated with the device-specific driver portion in a table using the device identifier (device type value), comparing versions associated with functions of the device-specific driver portion to versions expected (device mapping table) through an application program interface (device mapper operations). See col. 13, lines 60-66, col. 15, lines 14-45, col. 16, line 50 – col. 17, line 59. Given the teaching of Schoening, one of ordinary skill in the art would have been motivated to include locating and comparing into Bondy as modified because this would have allowed new devices to be added without requiring revision of the applications (col. 3, lines 24-33).

As to claims 3, 4, 20, 21, 35, Bondy teaches the device identifier includes an application-specific integrated circuit identifier / a graphics chip identifier (Silicon Graphics Inc., GL, IBM graPHICGS, col. 4, lines 55-58)

As to claims 5, 6, 18, 19, Keller teaches device driver architecture, wherein a hardware-specific driver portion includes direct draw functions (DD 66), and direct 3D functions (68 including D3D; col. 7, lines 46-60)

As to claim 7, Bondy teaches calling a function to load a block of executable code in kernel mode memory (col. 5, line 62 – col. 6, line 6).

As to claims 10, 14, Bondy teaches the device-independent driver code includes two-dimensional graphics functions (2-D module 56).

As to claim 13, note discussion of claim 1, and note the equivalence of device-independent functions / device-independent driver code. Bondy further teaches device-independent functions are capable of supporting a plurality of different display devices (package 56 supports devices A, B, C, D represented by the respective adapters); a plurality of device-

Art Unit: 2194

specific driver portions (device specific code; col. 4, lines 27-42), each only capable of supporting a portion of the plurality of different display devices (device specific code 81-84 support devices A, B, C, D respectively). Note claim 1 for second function to load and for kernel mode memory.

As to claim 31, it is a program product claim of claim 13, thus not claim 13 for discussion.

As to claims 11 and 36, Schoening teaches device driver management, including locating a name associated with the device-specific driver portion in a table using the device identifier (device type value), comparing versions associated with functions of the device-specific driver portion to versions expected (device mapping table) through an application program interface (device mapper operations). See col. 16, line 50 – col. 17, line 59. Given the teaching of Schoening, one of ordinary skill in the art would have been motivated to include locating and comparing into Bondy as modified because this would have allowed new devices to be added without requiring revision of the applications (col. 3, lines 24-33).

Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bondy et al. (U.S. 5,491,813) in view of Keller et al. (U.S. 5,752,032) and Schoening et al. (U.S. 6,226,788) further in view of Shirakabe et al. (U.S. 5,136,709).

As to claim 16, Shirakabe teaches loading device drivers, including determining addresses associated with functions of the particular device-specific driver portion (col. 8, lines 27-53). Given the teaching of Shirakabe, one of ordinary skill in the art would have been

motivated to include determining addresses into Bondy as modified because this would have provided independent configuration of the driver and the kernel (col. 10, lines 20-29).

The ground of rejection for claim 16 has been clarified to fix the typographical in the final rejection.

(10) Response to Argument

A. Rejection of Claims 1, 13 and 31 under 35 U.S.C. § 103 (37 C.F.R. § 1,192(c)(8)(iv)):

1) The Schoening reference fails to disclose identifying a particular device-specific driver portion

In the Brief, Appellant argued in substance that Schoening reference fails to disclose or suggest identifying a device specific driver portion based on a comparison of versions associated with functions of the device specific portion to versions expected through an application program interface (Brief page 9, lines 19-22) because Schoening reference discloses identifying functions of an application program interface to be overridden (Brief page 10, lines 8-9). Appellant argued that the Schoening reference discloses “the Device Mapper is a table associated with a device or device type in a network that identifies Service Module Functions that are overridden for the associated device or device type”, “a service module is a set of classes derived from the FrameWork and FrontEnd packages that define the API, data model, database, and abstract functions that implement network device services”, thus, the system disclosed in the Schoening reference can determine which Device Mapper applies for a particular device version (Brief page

10, line 17 – page 11, line 15). Appellant further argued there is no disclosure or suggestion in the Schoening reference that the set of classes, or their functions, are device-specific driver portions for at least the reason that the Schoening reference fails to teach that they are used to control the operation of a specific peripheral device in any manner. Since Schoening reference states that the set of classes “define the API, data model, database, and abstract functions that implement network device services”, thus, accordingly, the Device Mapper identifies portions of an API and associated *abstract functions* of network device services, rather than *device specific driver portion* (Brief page 11, line 16 – page 12, 8).

Examiner respectfully disagrees with the Appellant’s arguments, Schoening teaches identifying a device specific driver portion (“a mechanism for automatic determination of currently supported devices 102 at start-up time, and automatic integration of device-specific overrides of Service Module Functions at start-up time” (emphasis added); col. 12, line 55-61), wherein the each Service Module Function is a Java class that is adapted to a particular device (col. 13, lines 31-32 and lines 40-44), and the Service Module Functions may be instantiated in a device-specific manner at runtime when a service is needed by an application (col. 17, line 55 – col. 18, line 4), based on a comparison of versions associated with functions of the device specific portion to versions expected through an application program interface (“a Device Mapper 1214 is associated with each device 102 or device type. Each Device Mapper ... identifies a Service Module Functions 76a-76n that is overridden ... device type”; col. 13, lines 60-64 and “When a device has more than one version, the versions are handled by subclasses of the parent class ... to C55XX”; col. 15, lines 3-13 and “a Device Mapper is located for the version portion of the OID”; col. 17, lines 4-5). Thus, the reference of Schoening teaches the

claimed limitation “identifying a device specific driver portion based on a comparison of versions associated with functions of the device specific portion to versions expected through an application program interface”.

2) The proposed combination of the Bondy, Keller and Schoening references does not disclose or suggest each and every limitation of the claims under appeal.

As to Appellant’s arguments regarding claim 13, the claim recites providing a third function to manipulate a processor to load a particular device-specific portion into kernel mode memory, wherein the particular device-specific driver portion is associated with the particular display device of the plurality of different display devices, and Schoening reference fails to disclose or suggest identifying or loading device-specific driver portions (page 12, lines 17-21), examiner respectfully disagrees. In the rejection of claim 13, which recites claim 1, examiner shows that Bondy teaches load a particular device-specific (device specific code 81 or 82; col. 4, lines 39-42), wherein the particular device-specific driver portion is associated with the particular display device of the plurality of different display devices (display adapters A, B, C, D, E; see Fig. 3 and col. 7, lines 9-30), and Kelly teaches loading the portion into kernel mode memory (“The shell module 72 is the initial component of the device driver 50 loaded into the memory ... startup”; col. 7, line 61 – col. 8, line 14 and “As each hardware interface module 76-88 is loaded into the memory 16”; col. 15, line 24-25), not Schoening as argued by the Appellant. Therefore, the arguments are not persuasive.

As to Appellant’s arguments regarding claim 31 that Schoening fails to teach identify a particular device-specific driver by locating a name associated with the particular device-specific

driver portion in a table using the device identifier (Brief page 12, line 22 – page 13, line 2), examiner respectfully disagrees. Schoening teaches identify a particular device-specific driver (Service Module Functions 76a-76n ... is a Java class that is adapted to a particular device; col. 13, lines 27-32 and lines 40-44) by locating a name associated with the particular device-specific driver portion in a table (a Device Mapper is associated ... device type; col. 13, lines 61-66) using the device identifier (sysObjectIDs, each of which references a unique type of device 102; col. 13, line 66 – col. 14, line 1 and device type identifier; col. 14, lines 60-62). Also refer to response in the section 1 above. Therefore, the arguments are not persuasive.

Appellant further argued that Bondy, Keller and Schoening references do not teach “device driver management” (Brief page 13, lines 3-4), examiner respectfully disagrees. The “device driver management” is not claimed in any of the claims 1, 13 or 31, therefore, the arguments are not persuasive.

B. Rejection of claim 16 under 35 U.S.C. § 103 (37 C.F.R. § 1,192(c)(8)(iv)):

Appellant argued in substance that the claim 16 is depended on claim 13, and claim 16 did not rely upon the Schoening reference to support a prima facie case of obviousness, therefore, the Final Action fails to establish a prima facie case of obviousness in support of claim 16 (Brief page 13, lines 11-21).

The ground of rejection for claim 16 has been clarified to correct a typographical mistake. Furthermore, in the Final Action, the rejection of claim 16 addressed the new limitation that is cited, thus, the rest of the claim 16 which is depended on claim 13 is inherently rejected as in claim 13.

Art Unit: 2194


(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Diem Cao

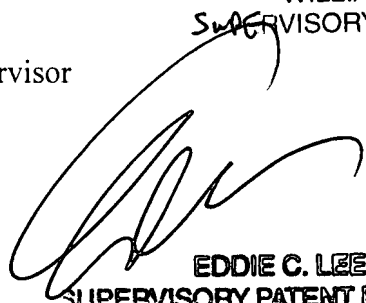


Conferees:

William Thomson, Supervisor Art Unit 2194


WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Eddie Lee, Supervisor


EDDIE C. LEE
SUPERVISORY PATENT EXAMINER